

CLAIMS

1. A method for realizing transactions with an existing protocol used by different system components communicating with each other, wherein each transaction contains update requests belonging together and each update request is generated by a first system component and transmitted to a second system component allowing access to a storage medium on which information to be updated is stored, said method being performed automatically and comprising the steps of:
 - (a) opening a transaction by adding a transaction control to an update request to create an extended update request supported by the protocol securing exchange of update requests between said system components;
 - (b) generating a transaction identifier for said transaction supported by said protocol;
 - (c) adding said transaction identifier to each update request belonging to said opened transaction; and
 - (d) automatically closing said transaction by using a transaction control supported by said protocol.
2. A method according to claim 1, wherein said first system component is installed on a client system and said second system component is installed on a server system.
3. A method according to claim 2, wherein said transaction identifier is generated by said server system.
4. A method according to claim 2, wherein said transaction identifier is generated by said client system.

5. A method according to claim 2, wherein said transaction identifier is sent from said server system to said client system.
6. A method according to claim 1, wherein step (a) is performed with a first update request of a sequence of update requests forming a transaction.
7. A method according to claim 1, wherein said update request is a delete, modify or add request.
8. A method according to claim 1, wherein said transaction control for opening a transaction according to step (a) has a value indicating a new transaction.
9. A method according to claim 1, wherein said step (d) is performed with a last update request of a sequence of update requests forming a transaction.
10. A method according to claim 1, wherein said transaction control for closing a transaction according to step (d) has a value indicating a commit or a rollback.
11. A method according to claim 1, wherein all extended update requests belonging to a transaction are stored to a queue and performed as a whole when an update request containing a transaction control having a value indicating a commit is received.
12. A method according to claim 1, wherein each individual extended update request belonging to a transaction is performed immediately after data to be updated has been stored on a nonvolatile storage medium.
13. A method according to claim 1, wherein said protocol is LDAP V3.

14. A method according to claim 13, wherein said transaction control and said transaction identifier each comprise a control name and a control value, wherein said name comprise a unique LDAP object identifier and said value comprises binary data.
- 5 15. A method according to claim 1, wherein said information to be updated is stored in a directory information tree.
16. A method according to claim 1, wherein an update request containing neither a transaction control nor a transaction identifier is performed as a single atomic request.
- 10 17. A method according to claim 1, wherein an update request with a transaction control with a syntactically invalid value is not performed.
18. A method according to claim 1, wherein all update requests identified by a transaction identifier are performed, a transaction control having a value indicating a commit and said transaction identifier are added to the response, and the transaction identified by said transaction identifier is closed if an update request containing a transaction control having a value indicating a commit and a valid transaction identifier corresponding to said transaction identifier is received by said server system.
- 20 19. A method according to claim 1, wherein all effects caused by said update requests belonging to the transaction identified by said transaction identifier are rolled back, said transaction is closed, and a response is extended with a transaction control having a value indicating a rollback and with said transaction identifier if said update request cannot be performed successfully.
- 25 20. A method according to claim 1, wherein a new transaction is opened when an update request does not contain a transaction identifier and contains a transaction control with a value indicating a new transaction.
- 30

21. A method according to claim 1 wherein all effects caused by said update requests are rolled back, the transaction identified by said transaction identifier is closed, and a response is extended with a transaction control having a value indicating a new transaction but not with said transaction identifier if the first request of a transaction cannot be performed successfully.

22. A computer program product containing parts of software code for executing the method in accordance with claim 1 when said software code is running on a digital computer.

23. A system for updating information comprising:

a client system containing a client program for adding transaction control information and a transaction identifier to update requests according to the claim 1; and

a server system containing a server program for generating a transaction identifier according to claim 1 and accessing or updating information; and

a data storage medium containing information to be accessed or updated.

24. A system according to claim 23 wherein said client program is an LDAP client program, said server program is an LDAP server program, and said information to be updated or accessed is stored in a directory information tree.

25. A system according to claim 24, further comprising:

a backend for managing a subtree of said directory information tree containing information to be updated; and

a component which is part of said server program for routing an update request to said backend managing said subtree of said directory information tree containing the information to be updated.

5 26. A system comprising:

a client data processing system; and

10 a client program installed on said system comprising at least a component for adding transaction control information and a transaction identifier to update requests according to claim 1.

15 27. A system comprising:

a server data processing system; and

20 a server program installed on said system comprising at least a component for generating a transaction identifier according to claim 1 and a component for accessing or updating data.